www.cis.upenn.edu/~bcpierce/courses/670Fall04/GreatWorksInPL.shtml

Great Works in Programming Languages

Collected by Benjamin C. Pierce

In September, 2004, I posted a query to the Types list asking people to name the five most important papers ever written in the area of programming languages. This page collects the responses I received. (A few are missing because I am still tracking down bibliographic information.)

Many thanks to Frank Atanassow, David Benson, Nick Benton, Karl Crary, Olivier Danvy, Mariangiola Dezani, Dan Friedman, Alwyn Goodloe, Pieter Hartel, Michael Hicks, Robert Irwin, Luis Lamb, Rod Moten, Rishiyur Nikhil, Tobias Nipkow, Jens Palsberg, and John Reynolds for contributing.

Additional suggestions are welcome. (Bibtex format preferred!)

-- BCP

The greatest of the great (mentioned by many people):

C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576-580 and 583, October 1969.
[bib]

Peter J. Landin. The next 700 programming languages. *Communications of the ACM*, 9(3):157-166, March 1966.
[bib]

Robin Milner. A theory of type polymorphism in programming. *Journal of Computer and System Sciences*, 17:348-375, August 1978.
[bib]

Gordon Plotkin. Call-by-name, call-by-value, and the λ-calculus. *Theoretical Computer Science*, 1:125-159, 1975.
[bib]

John C. Reynolds. Towards a theory of type structure. In *Colloque sur la Programmation, Paris, France*, volume 19 of *Lecture Notes in Computer Science*, pages 408-425. Springer-Verlag, 1974.
[bib]

Pretty great works (mentioned by multiple people):

Luca Cardelli. A semantics of multiple inheritance. In G. Kahn, D. MacQueen, and G. Plotkin, editors, *Semantics of Data Types*, volume 173 of *Lecture Notes in Computer Science*, pages 51-67. Springer-Verlag, 1984. Full version in Information and Computation, 76(2/3):138-164, 1988.
[bib]

Luis Damas and Robin Milner. Principal type schemes for functional programs. In *ACM Symposium on Principles of Programming Languages (POPL), Albuquerque, New Mexico*, pages 207-212, 1982.
[bib]

Edsger W. Dijkstra. Recursive programming. In Saul Rosen, editor, *Programming Systems and Languages*, chapter 3C, pages 221-227. McGraw-Hill, New York, 1960.
[bib]

Edsger W. Dijkstra. Go to statement considered harmful. 11(3):147-148, March 1968.
[bib]

William A. Howard. The formulas-as-types notion of construction. In J. P. Seldin and J. R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*, pages 479-490. Academic Press, 1980. Reprint of 1969 article.
[bib]

Robert Kowalski. Predicate logic as programming language. In *IFIP Congress*, pages 569-574, 1974. Reprinted in Computers for Artificial Intelligence Applications, (eds. Wah, B. and Li, G.-J.), IEEE Computer Society Press, Los Angeles, 1986, pp. 68-73.
[bib]

Peter J. Landin. The mechanical evaluation of expressions. *Computer Journal*, 6(4):308-320, January 1964.
[bib]

John McCarthy. Recursive functions of symbolic expressions and their computation by machine, part I. *Communications of the ACM*, 3(4):184-195, April 1960.
[bib]

Eugenio Moggi. Computational lambda-calculus and monads. In *IEEE Symposium on Logic in Computer Science (LICS), Asilomar, California*, pages 14-23, June 1989. Full version, titled *Notions of Computation and Monads*, in Information and Computation, 93(1), pp. 55-92, 1991.
[bib]

Greg Morrisett, David Walker, Karl Crary, and Neal Glew. From System-F to typed assembly language. *ACM Transactions on Programming Languages and Systems*, 21(3):527-568, May 1999.
[bib]

George C. Necula. Proof-carrying code. In *ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), Paris, France*, pages 106-119, January 1997.
[bib]

Gordon D. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5:223-255, 1977.
[bib]

Gordon D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Computer Science Department, Aarhus University, Aarhus, Denmark, September 1981.
[bib]

Jr. Guy Lewis Steele. RABBIT: A compiler for SCHEME. Technical Report AITR-474, MIT Artificial Intelligence Laboratory, May 6 1978.
[bib | .pdf]

A great many great works (mentioned at least once):

C. Boehm and G. Jacopini. Flow diagrams, Turing machines, and languages with only two formation rules. *Communications of the ACM*, 9(5):366-371, 1966.
[bib]

Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56-68, 1940.

[bib]
Alonzo Church. *The Calculi of Lambda Conversion*. Princeton University Press, 1941.

[bib]
Haskell B. Curry and Robert Feys. *Combinatory Logic*, volume 1. North Holland, 1958. Second edition, 1968.

[bib]
O. J. Dahl and K. Nygaard. SIMULA-An ALGOL-based simulation language. *Communications of the ACM*, 9(9):671-678, September 1966.

[bib]
Matthias Felleisen. On the expressive power of programming languages. *Science of Computer Programming*, 17(1-3):35-75, December 1991.

[bib | www]
Andrzej Filinski. Representing layered monads. In *ACM Symposium on Principles of Programming Languages (POPL), San Antonio, Texas*, pages 175-188, New York, NY, 1999.

[bib | .html]
Robert W. Floyd. Assigning meanings to programs. In J. T. Schwartz, editor, *Mathematical Aspects of Computer Science*, volume 19 of *Proceedings of Symposia in Applied Mathematics*, pages 19-32, Providence, Rhode Island, 1967. American Mathematical Society.

[bib]
Michael J. Gordon, Robin Milner, F. Lockwood Morris, Malcolm Newey, and Christopher P. Wadsworth. A metalanguage for interactive proof in LCF. Internal Report CSR-16-77, University of Edinburgh, Edinburgh, Scotland, September 1977.

[bib]
C. A. R. Hoare. Proof of a program: FIND. *Communications of the ACM*, 14(1):39-45, January 1971.

[bib]
C. A. R. Hoare. Communicating sequential processes. *Communications of the ACM*, 21(8):666-677, August 1978. Reprinted in ``Distributed Computing: Concepts and Implementations" edited by McEntire, O'Reilly and Larson, IEEE, 1984.

[bib]
Thomas Johnsson. Lambda lifting: transforming programs to recursive equations. In *Functional programming languages and computer architecture. Proc. of a conference (Nancy, France, Sept. 1985)*, pages 190-203, New York, NY, USA, 1985. Springer-Verlag Inc.

[bib | .html]
Neil D. Jones, Carsten K. Gomard, and Peter Sestoft. *Partial Evaluation and Automatic Program Generation*. Prentice Hall International, Hemel Hempstead, Hertfordshire, UK, 1993.

[bib]
G. Kahn. The semantics of a simple language for parallel programming. In J. L. Rosenfeld, editor, *Information processing*, pages 471-475, Stockholm, Sweden, Aug 1974. North Holland, Amsterdam.

[bib]
Gilles Kahn. Natural semantics. In Franz-Josef Brandenburg, Guy Vidal-Naquet, and Martin Wirsing, editors, *Proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 247 of *Lecture Notes in Computer Science*, pages 22-39. Springer-Verlag, 1987.

[bib]
Donald E. Knuth. Structured programming with go to statements. *Computing Surveys*, 6(4):261-301, 1974.

www.cis.upenn.edu/~bcpierce/courses/670Fall04/GreatWorksInPL.shtml

[bib]
Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer-Verlag, 1980.

[bib]
Peter Naur et al. Revised report on the algorithmic language ALGOL 60. *Communications of the ACM*, 6(1):1-17, January 1963.

[bib]
John C. Reynolds. Types, abstraction, and parametric polymorphism. In R. E. A. Mason, editor, *Information Processing 83, Paris, France*, pages 513-523. Elsevier, 1983.

[bib]
John C. Reynolds. Definitional interpreters for higher-order programming languages. *Higher-Order and Symbolic Computation*, 11(4):363-397, 1998. Reprinted from the proceedings of the 25th ACM National Conference (1972), with a foreword.

[bib]
Dana Scott. Continuous lattices. In F. W. Lawvere, editor, *Toposes, Algebraic Geometry, and Logic*, number 274 in Lecture Notes in Mathematics, pages 97-136. Springer-Verlag, 1972.

[bib]
Dana Scott and Christopher Strachey. Toward a mathematical semantics for computer languages. Programming Research Group Technical Monograph PRG-6, Oxford Univ. Computing Lab., 1971.

[bib]
Dana S. Scott. Outline of a mathematical theory of computation. Technical Monograph PRG-2, Oxford University Computing Laboratory, Oxford, England, November 1970.

[bib]
Brian Cantwell Smith. Reflection and semantics in lisp. In *ACM Symposium on Principles of Programming Languages (POPL), Salt Lake City, Utah*, pages 23-39, January 1984.

[bib]
Christopher Strachey. Towards a formal semantics. In *Formal Language Description Languages for Computer Programming*, pages 198-220. North Holland, 1966.

[bib]
Christopher Strachey. Fundamental concepts in programming languages. Lecture Notes, International Summer School in Computer Programming, Copenhagen, August 1967. Reprinted in *Higher-Order and Symbolic Computation*, 13(1/2), pp. 1-49, 2000.

[bib]
Christopher Strachey and Christopher P. Wadsworth. Continuations: A mathematical semantics for handling full jumps. Programming Research Group Technical Monograph PRG-11, Oxford Univ. Computing Lab., 1974. Reprinted in *Higher-Order and Symbolic Computation*, vol. 13 (2000), pp. 135-152.

[bib]
D. A. Turner. A new implementation technique for applicative languages. *Software - Practice and Experience*, 9(1):31-49, Jan 1979.

[bib]
David Ungar and Randall B. Smith. Self: The power of simplicity. *Lisp and Symbolic Computation*, 4(3):187-205, 1991.

[bib]
N. Wirth. The programming language Pascal (revised report). Technical report 5, Dept. Informatik, Inst. Für Computersysteme, ETH Zürich, Zürich, Switzerland, Jul 1973.

www.cis.upenn.edu/~bcpierce/courses/670Fall04/GreatWorksInPL.shtml

[bib]

Niklaus Wirth and C. A. R. Hoare. A contribution to the development of ALGOL. *Communications of the ACM*, 9(6):413-432, June 1966.

[bib]

Andrew K. Wright and Matthias Felleisen. A syntactic approach to type soundness. *Information and Computation*, 115(1):38-94, November 1994.

[bib]